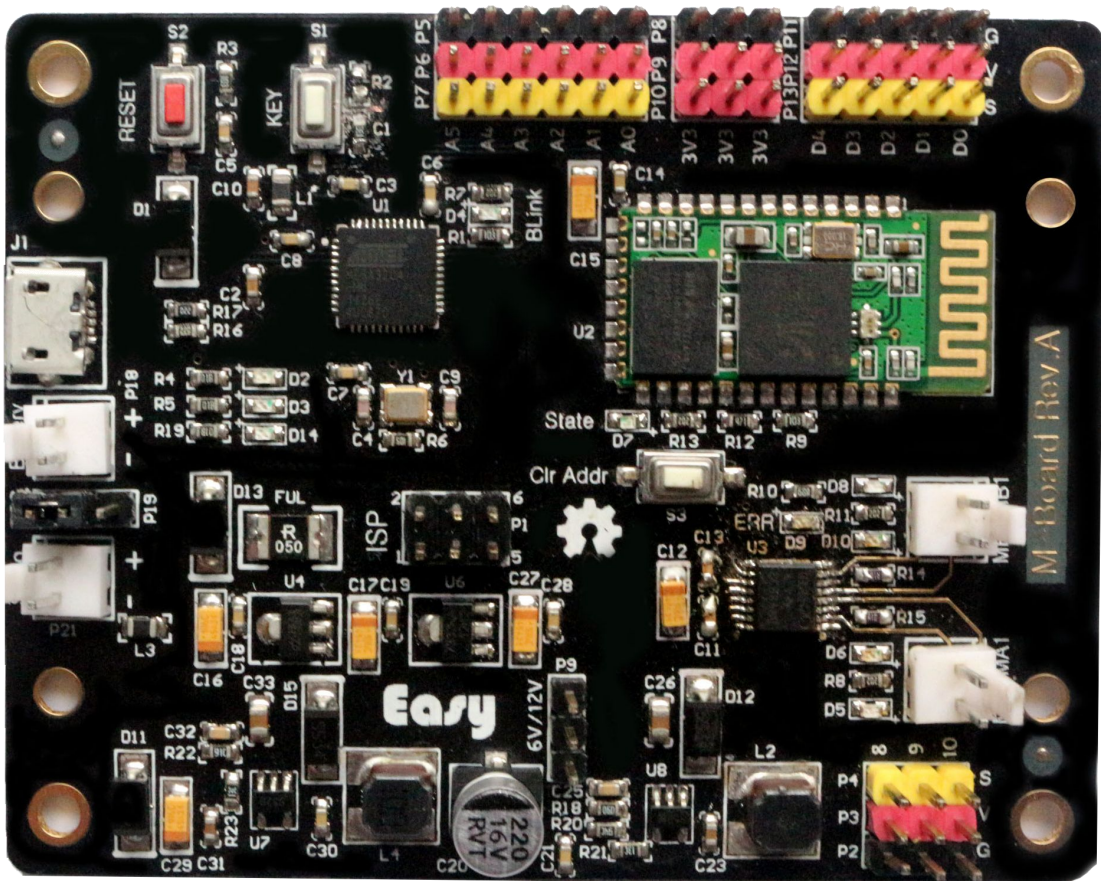
快速上手手册

M-Board Rev.A

概述

M-Board 能帮你快速搭建蓝牙控制的电机类应用，激发你的创意。M-Board 兼容 Arduino Leonardo，配合我们的驱动库，你能方便的控制电机和蓝牙。



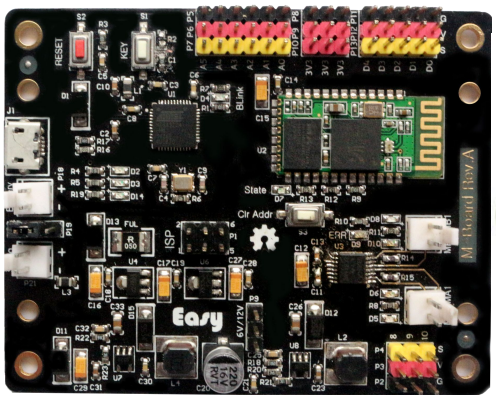
目 录

1 技术规格

- 2 路直流电机接口，双通道 H 桥电流控制电机驱动器控制
- 电机驱动芯片，具有过流保护、短路保护、欠压闭锁和过热保护功能。
- 可设置电机驱动电压 6V 或者 12V
- 支持电机正反转和 PWM 调速
- 板载升压电路适合电池供电
- 电源输入范围 3-9V DC
- 蓝牙 2.0 或者蓝牙 4.0 BLE，支持 AT 命令
- 3 路舵机接口
- 兼容 Arduino Leonardo
- 尺寸：80x63mm

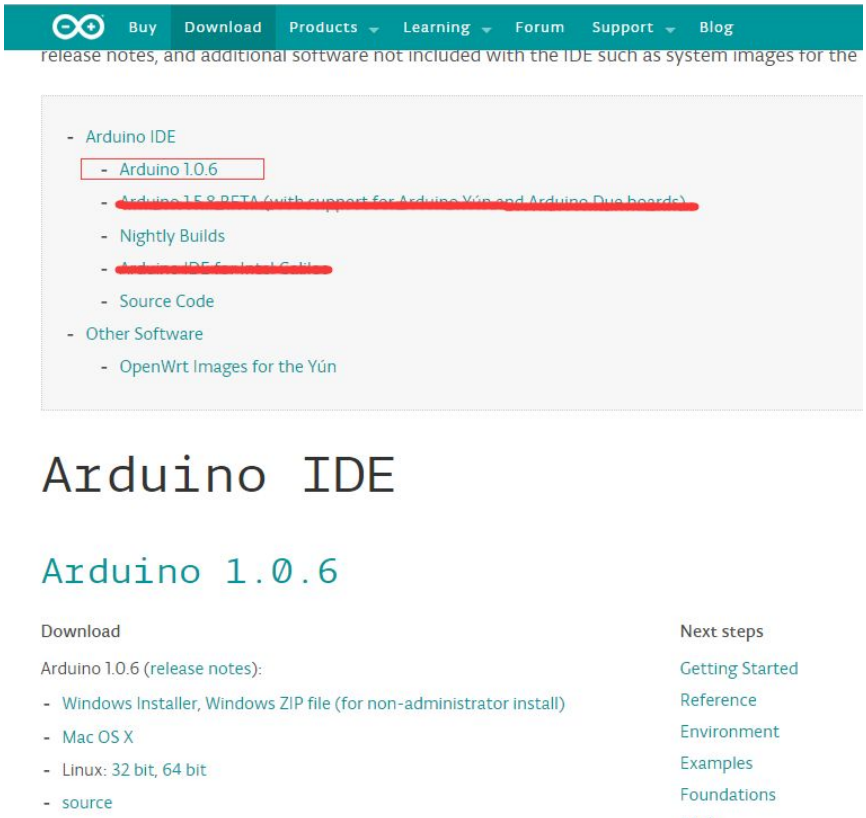
2 快速上手体验

除了 M-Board，你需要一台能上网的电脑和一根 Micro USB 数据线。



2.1 Arduino IDE 安装

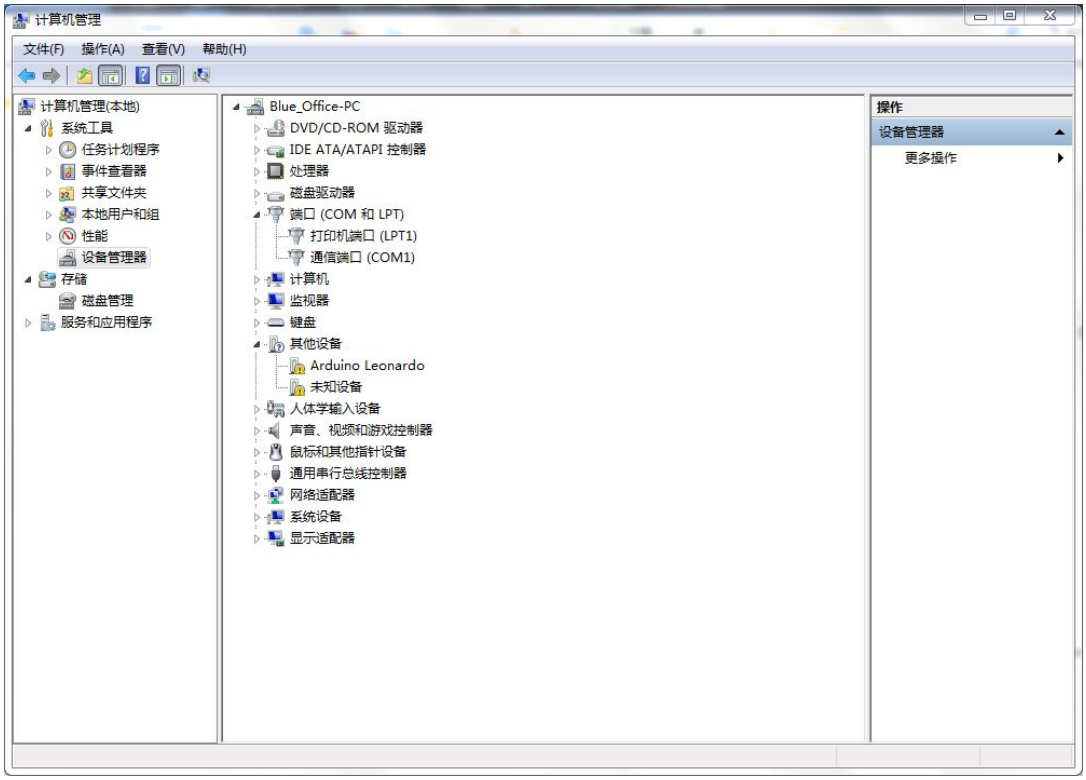
请到 Arduino 官方网站下载适合你系统的最新的安装文件。注意 M-Board 兼容 Arduino Leonardo。



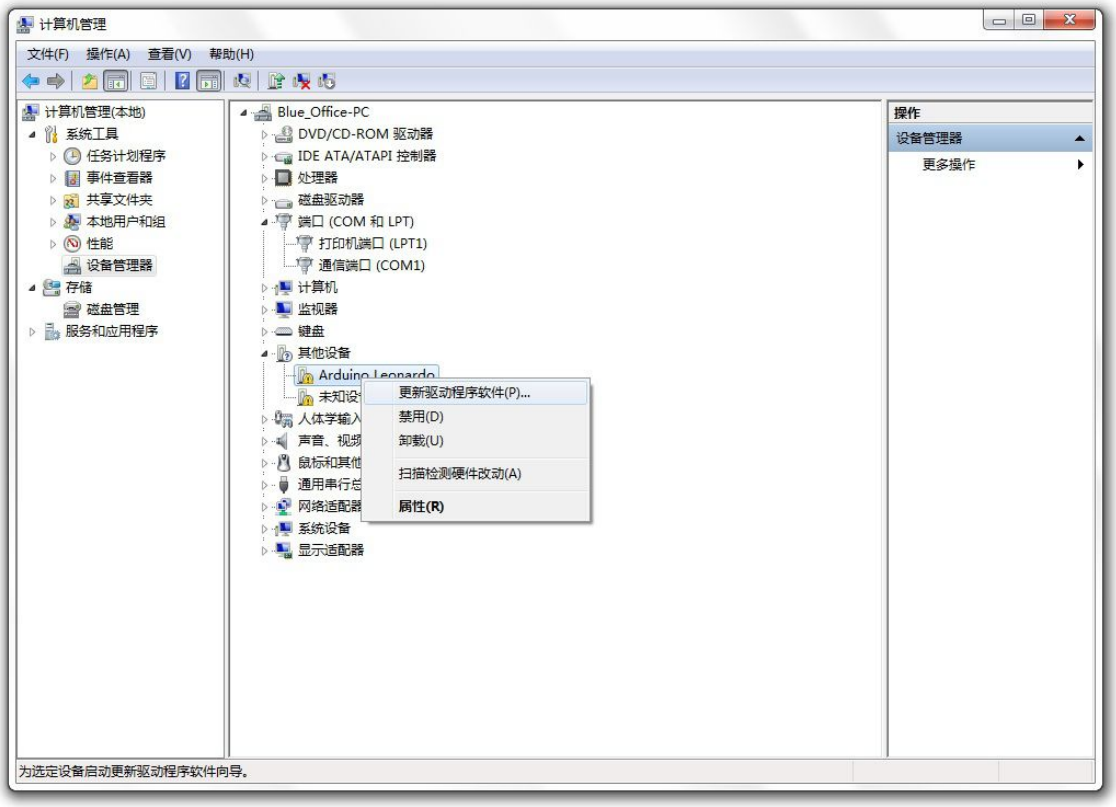
2.2 Leonardo 驱动安装

这里以 WIN7 系统下演示操作，其他系统请到官网查看相应的安装方法。

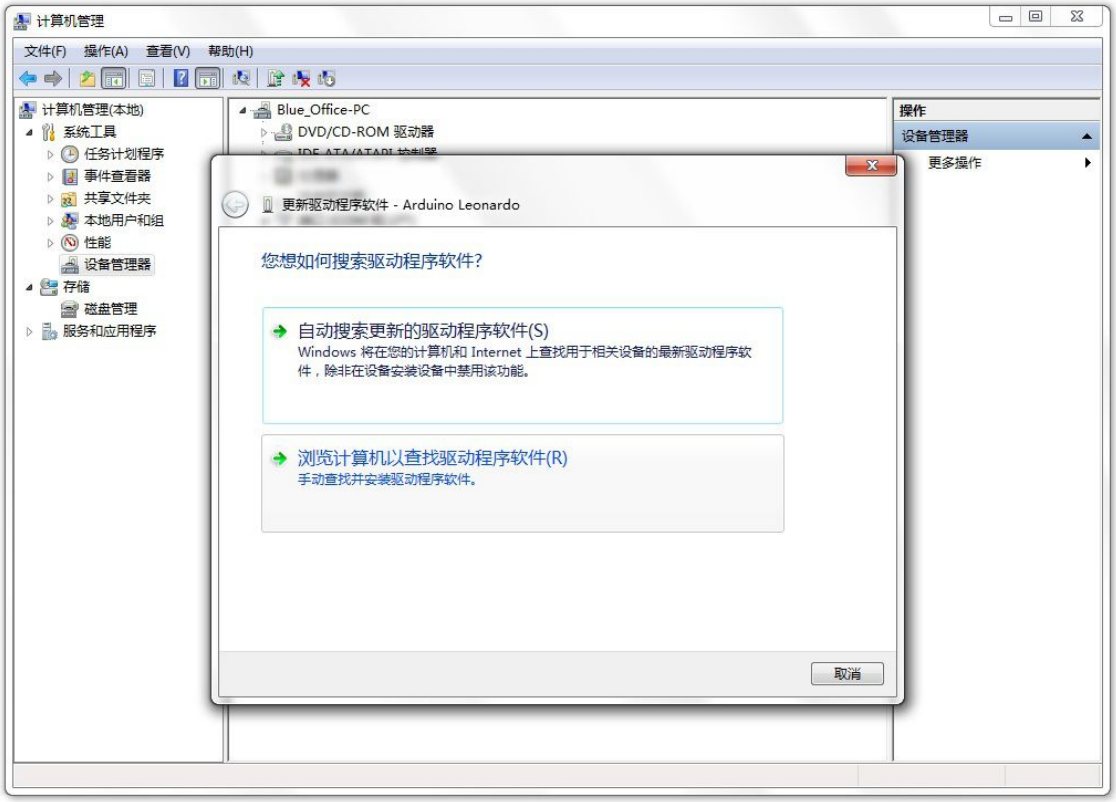
插上 USB 数据线后，在电脑的设备管理器中会看到其他设备里显示带黄色叹号的 Arduino Leonardo。



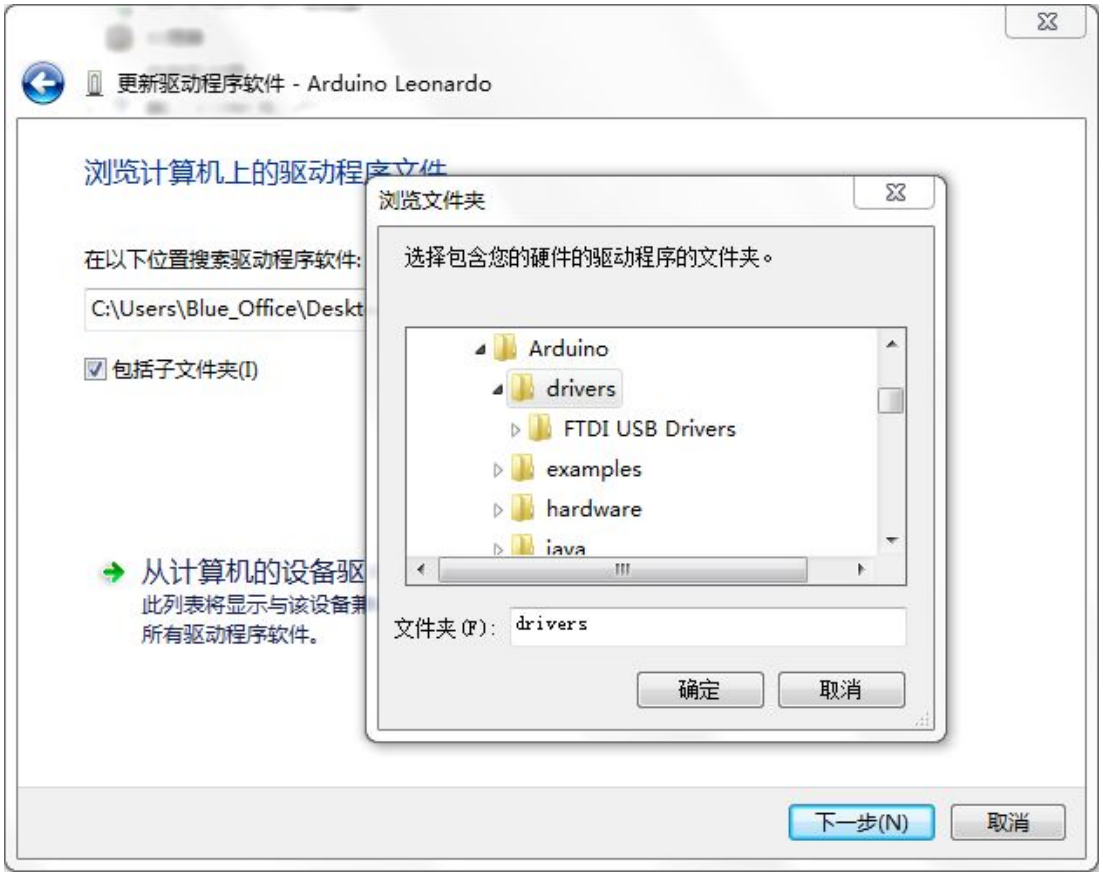
选择后右键，点击更新驱动程序软件

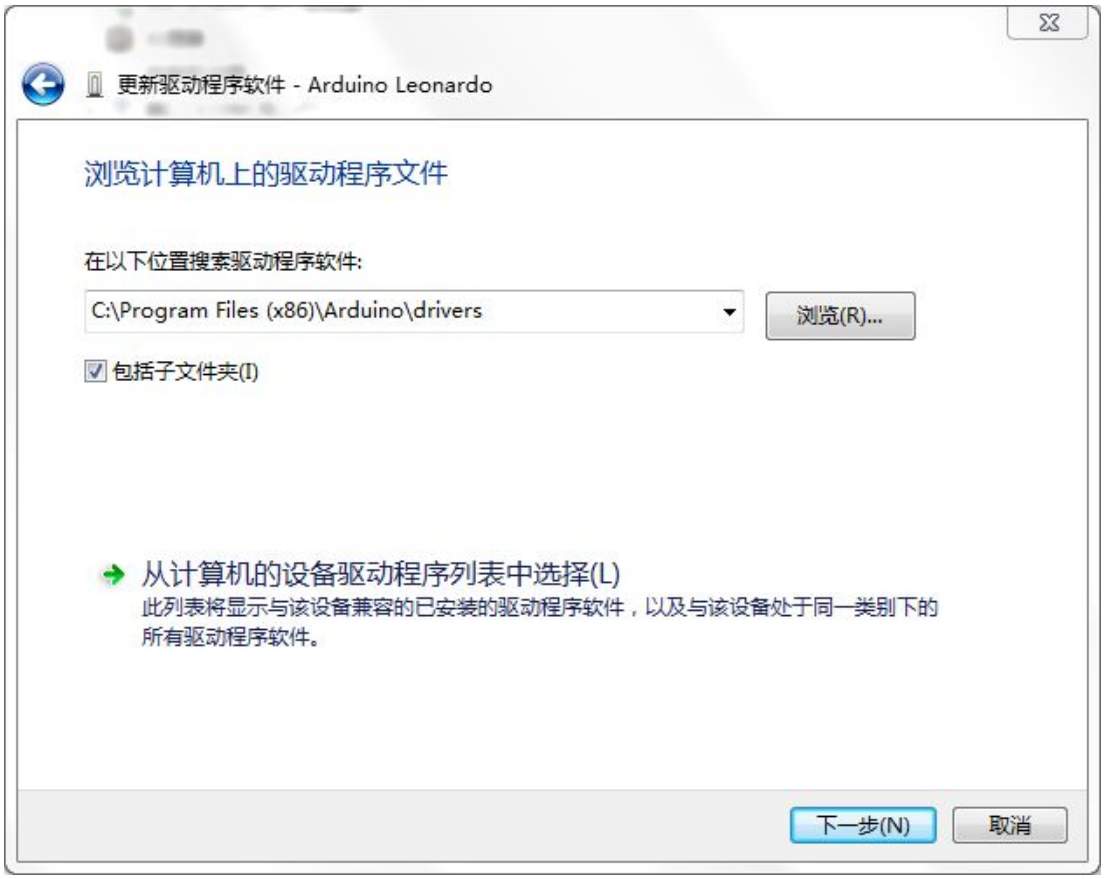


然后，选择浏览计算机以查找驱动程序软件



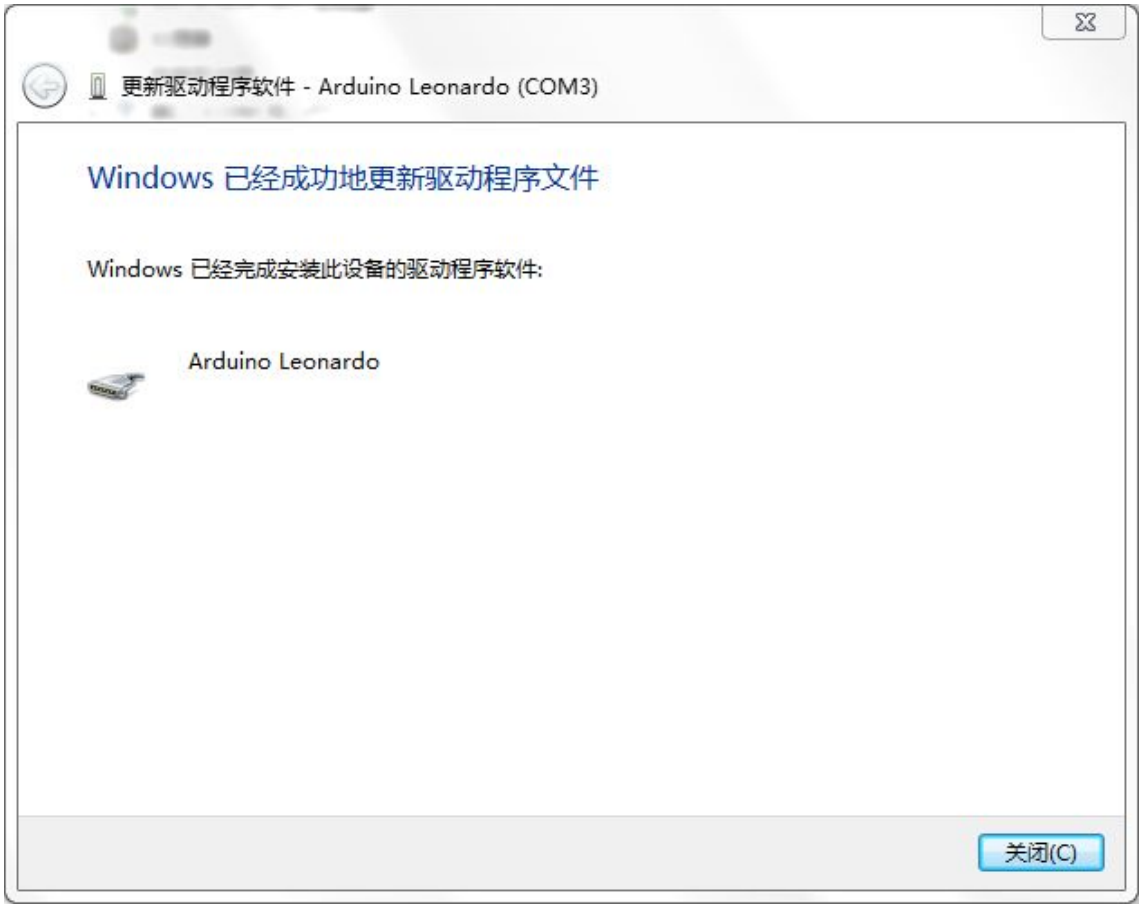
找到 Arduino 的安装目录下的 drivers 文件夹





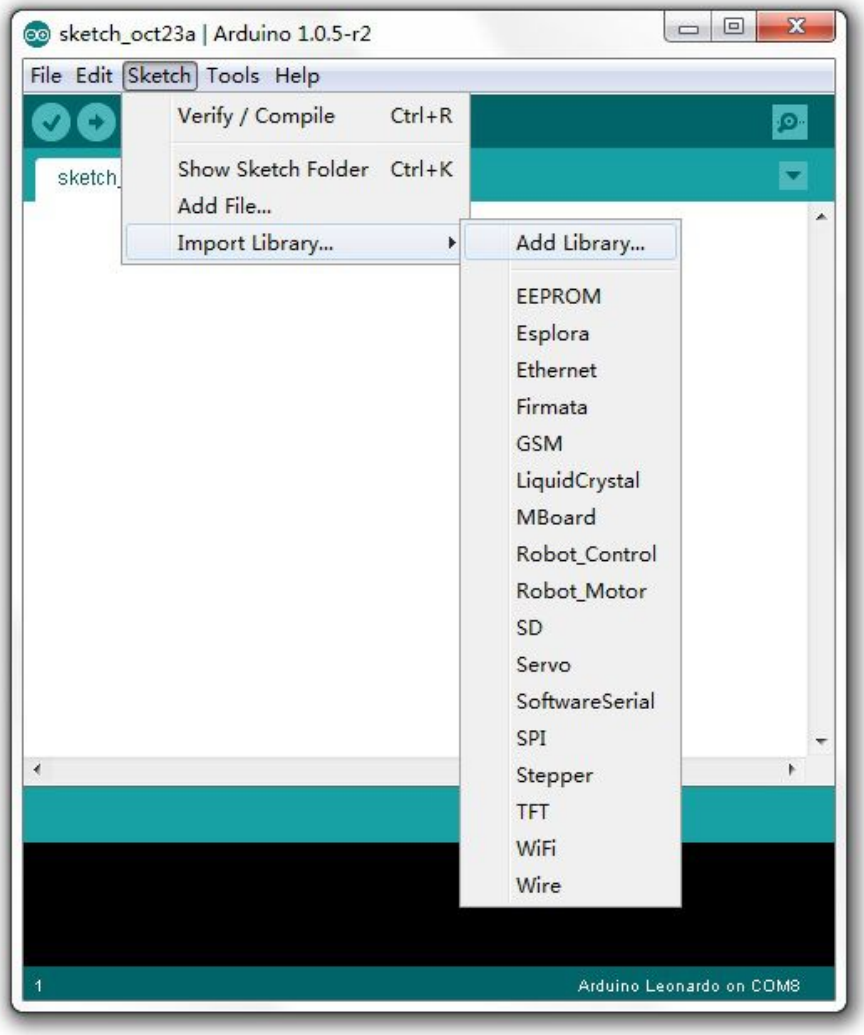
点击下一步后，会弹出安全警告，点击安装就行。

稍等一会，就会提示驱动安装完成，并给你分配了相应的端口号，后面会用到。



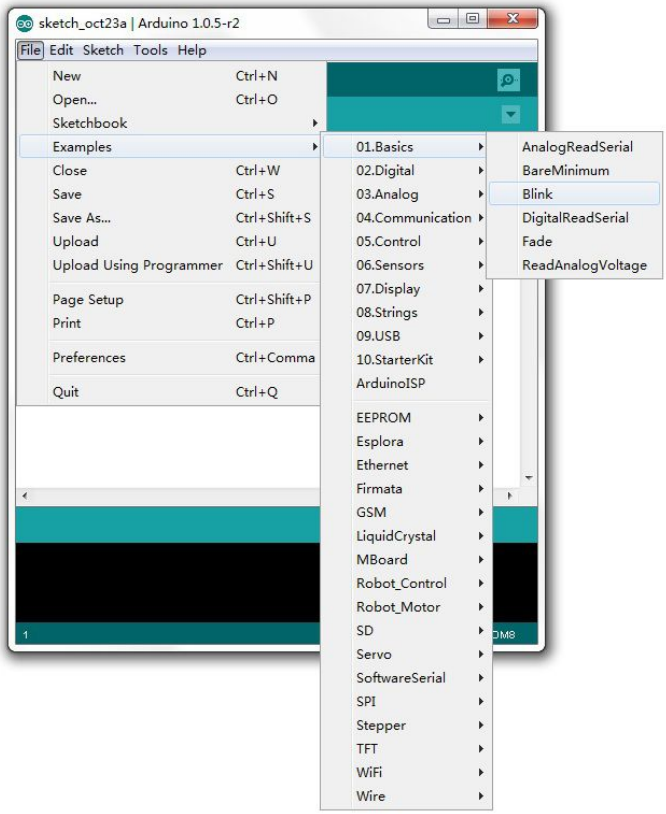
2.3 驱动库安装

下载 M-board Arduino 驱动库，解压后，打开 Arduino IDE，按下图操作，添加 Mboard 文件夹。

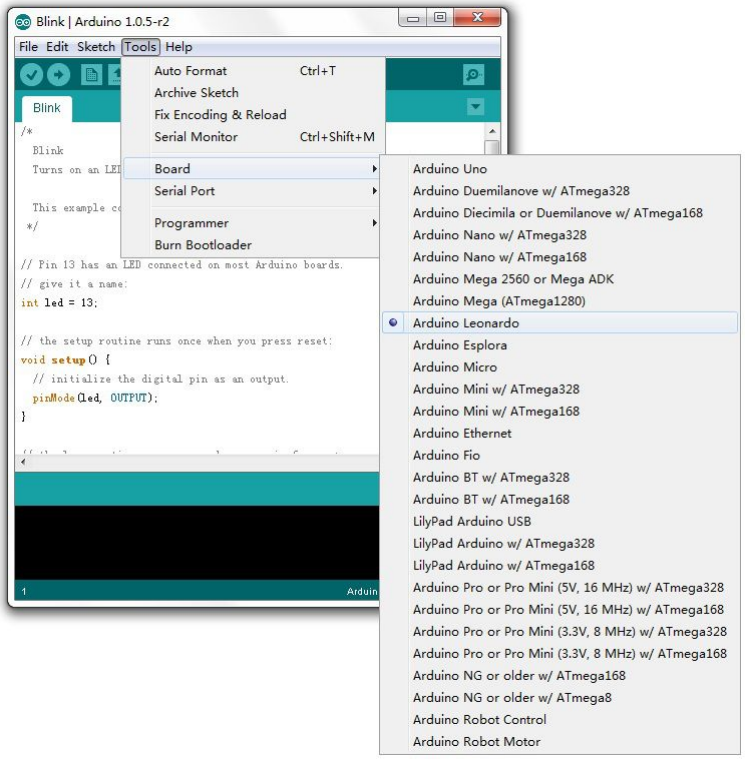


2.4 烧写 LED 闪烁程序

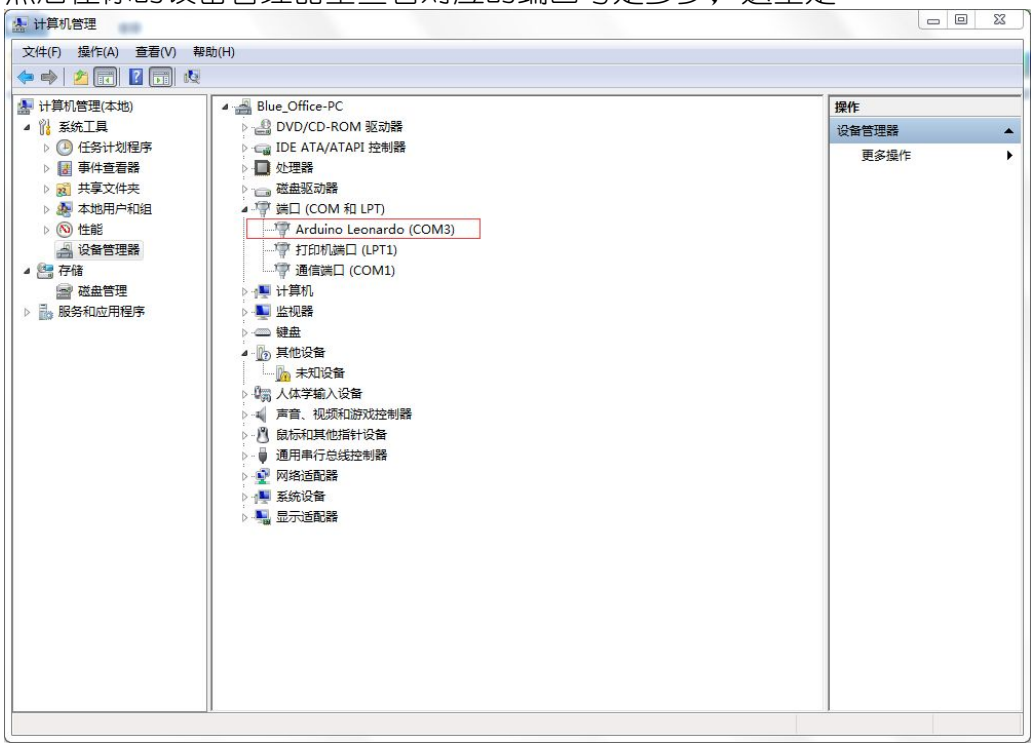
按照如下图所示，打开 Blink 程序。



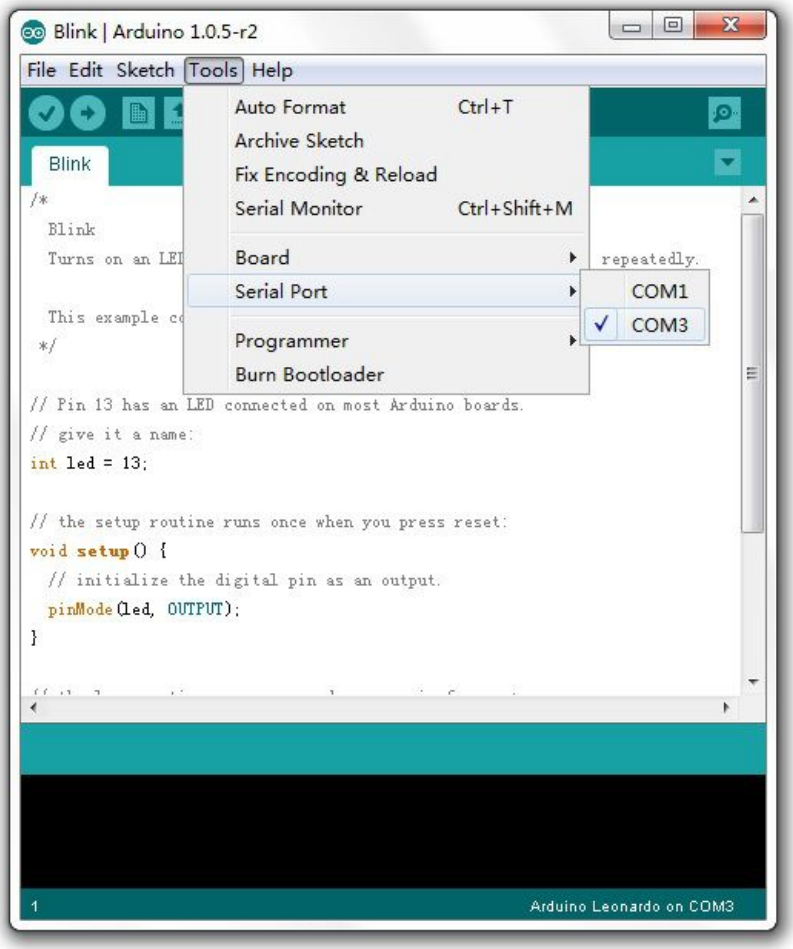
选择对应的电路板型号，我们选择 Arduino Leonardo。



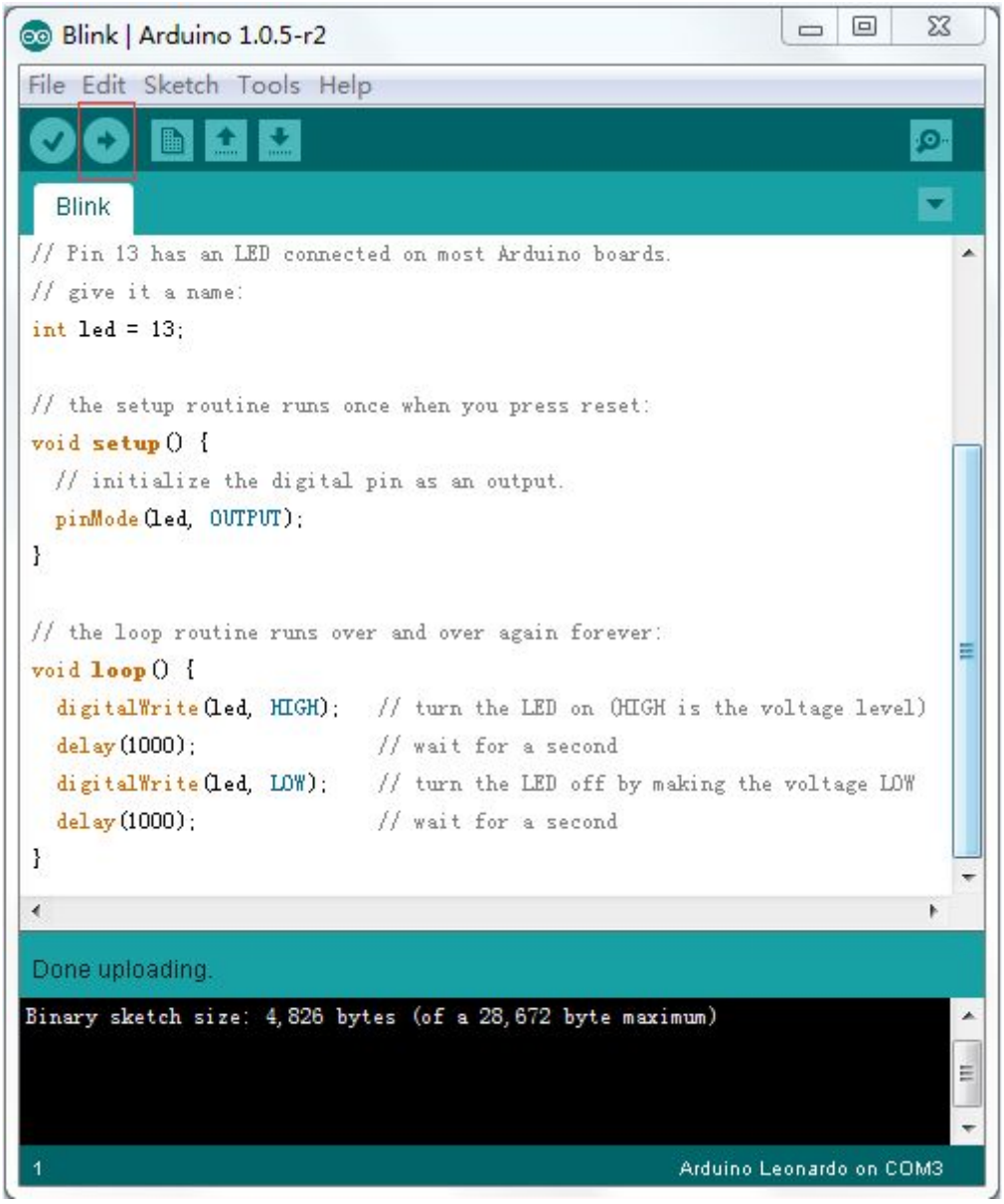
然后在你的设备管理器里查看对应的端口号是多少，这里是 COM3



于是，我们在 Serial Port 这里选择 COM3



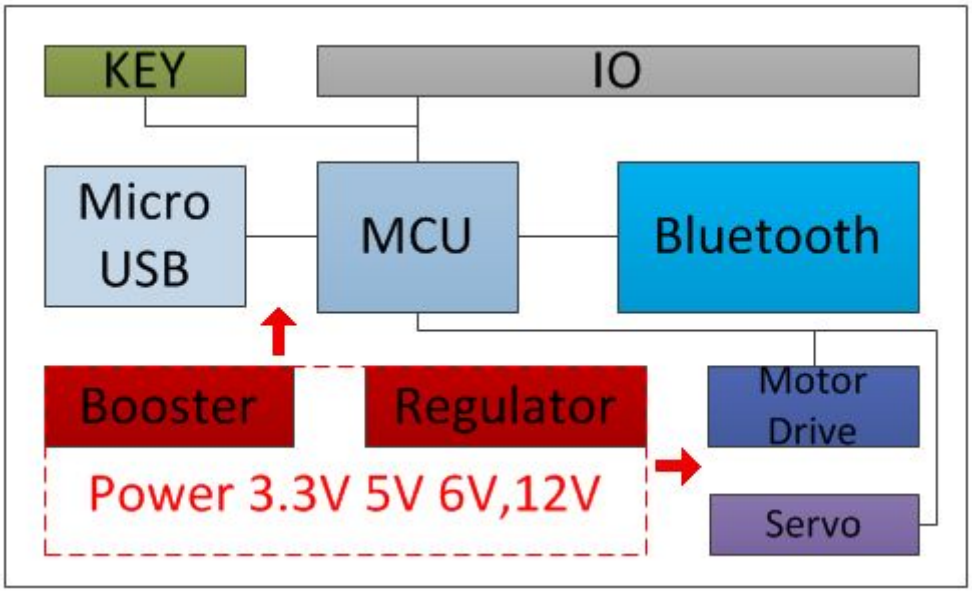
点击左上方的 Upload 按钮，等下面的状态栏显示 Done uploading 时，程序已经下载到你的 M-Board 了，你会看到你的 M-Board 上的 Blink LED 灯会每隔一秒的交替亮灭。



希望你看到这里，你能基本明白整个流程，接下来更多高级的应用，请看下面的说明。

3 硬件

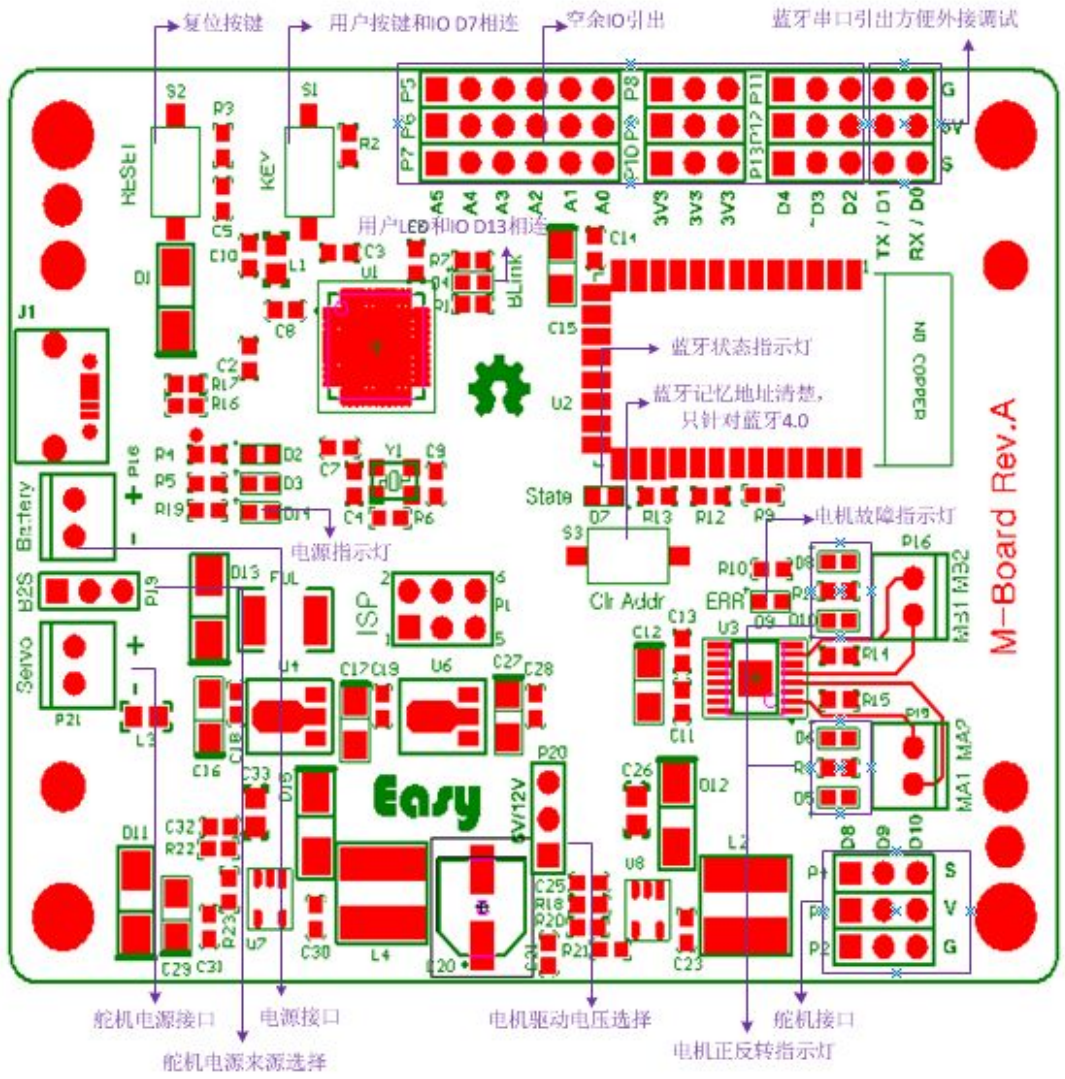
3.1 功能框图


KEY
IO
Micro USB
MCU
Bluetooth
Booster
Regulator
Motor Drive
Power 3.3V 5V 6V,12V
Servo

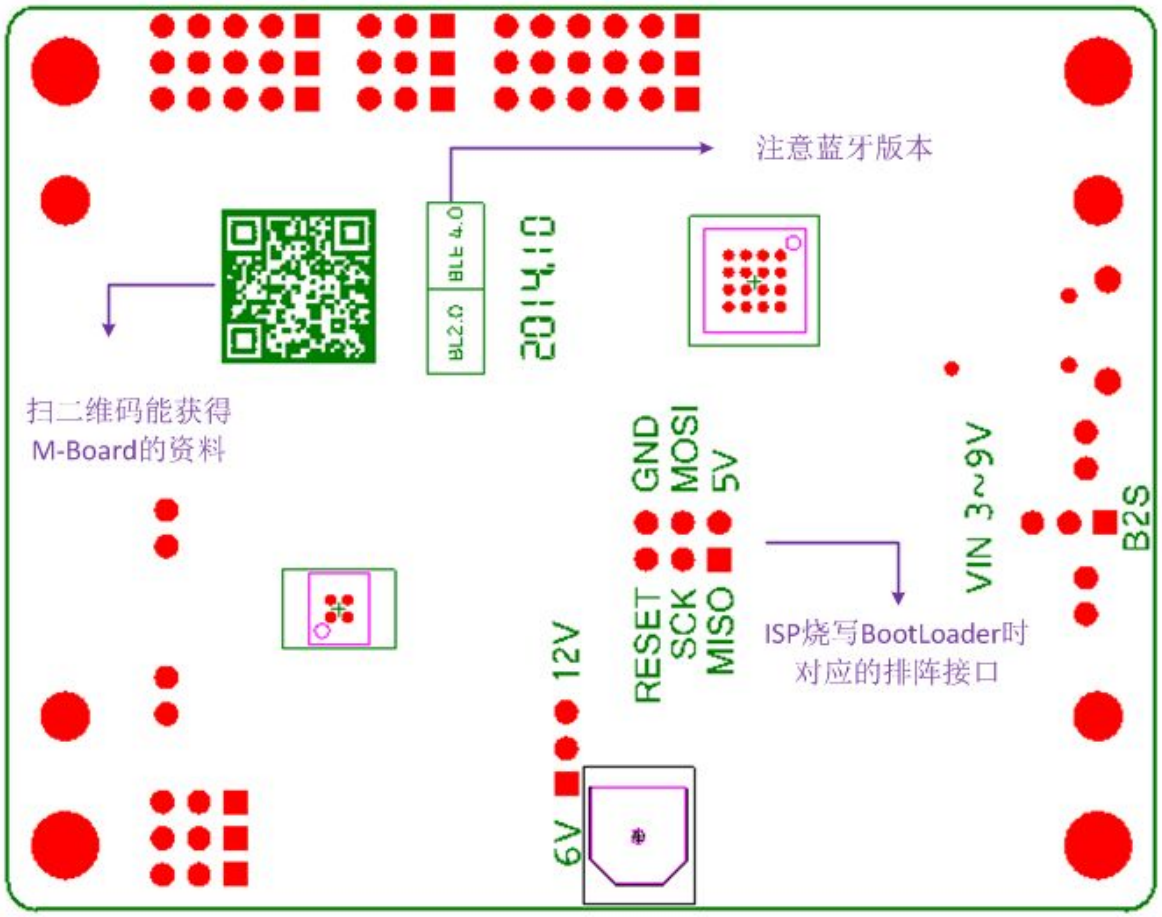
3.2 引脚分配

IO 引脚	说明
A0~A5	端口引出
D0	蓝牙接收
D1	蓝牙发送
D2, D3, D4	端口引出
D7	按键
D13	LED
D5	电机驱动芯片控制端口 AIN2
D6	电机驱动芯片控制端口 BIN1
D11	电机驱动芯片控制端口 AIN1
D12	电机驱动芯片控制端口 BIN2
D8	舵机 1 信号线
D9	舵机 2 信号线
D10	舵机 3 信号线

3.3 正面布局


复位按键
用户按键和IO D7相连
空余IO引出
蓝牙串口引出方便外接调试
用户LED和IO D13相连
蓝牙状态指示灯
蓝牙记忆地址清楚，
只针对蓝牙4.0
电源指示灯
电机故障指示灯
舵机电源接口
电源接口
电机驱动电压选择
舵机接口
舵机电源来源选择
电机正反转指示灯
M-Board Rev.A
Easy

3.4 背面布局


注意蓝牙版本
BLE 4.0
BL2.0
2014.10
扫二维码能获得
M-Board的资料
GND
MOSI
5V
RESET
SCK
MISO
ISP烧写BootLoader时
对应的排阵接口
VIN 3~9V
B2S
12V
6V

3.5 电源供电

电机，舵机和单片机系统的供电都是独立分开的，所以在单独USB供电时，只能烧写程序等基本功能，不能驱动电机和舵机这些大电流外设。

外接电池供电时，电压范围为6V~9V，需要AA电池3到6节，不能大于6节，否则会对电路板造成损坏。不同电池数量驱动能力不同。当需要驱动12V电机时，推荐使用6节AA电池；当需要驱动6V电机时，推荐使用4节AA电池。电池推荐使用镍氢电池或者动力锂电池。

舵机电源可以使用独立的电源，也可以和电机使用同一电源，主要要注意使用的舵机的电压范围，不然会对舵机造成损坏。

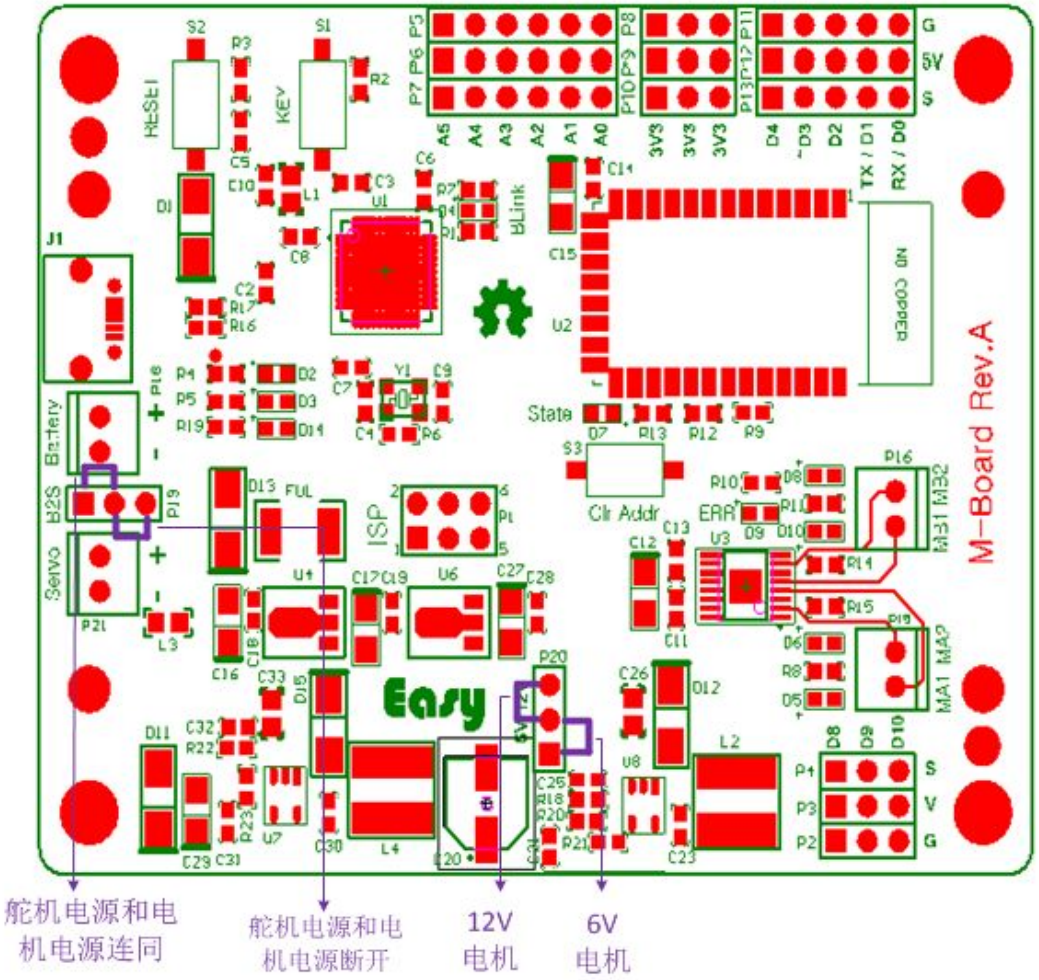
3.6 LED 灯说明

标识	颜色	功能	初始状态
D14	绿色	电源正常亮	亮
D2	黄色	接收过程中亮	灭
D3	黄色	发送过程中亮	灭
D4	绿色	可用户自定义	用户自定义
D7	红色	蓝牙未连接时闪烁，连上后常亮	闪烁
D9	红色	电机故障亮	灭
D8	绿色	电机正转亮	灭
D6	绿色	电机正转亮	灭
D10	黄色	电机反转亮	灭
D5	黄色	电机反转亮	灭

3.7 按键

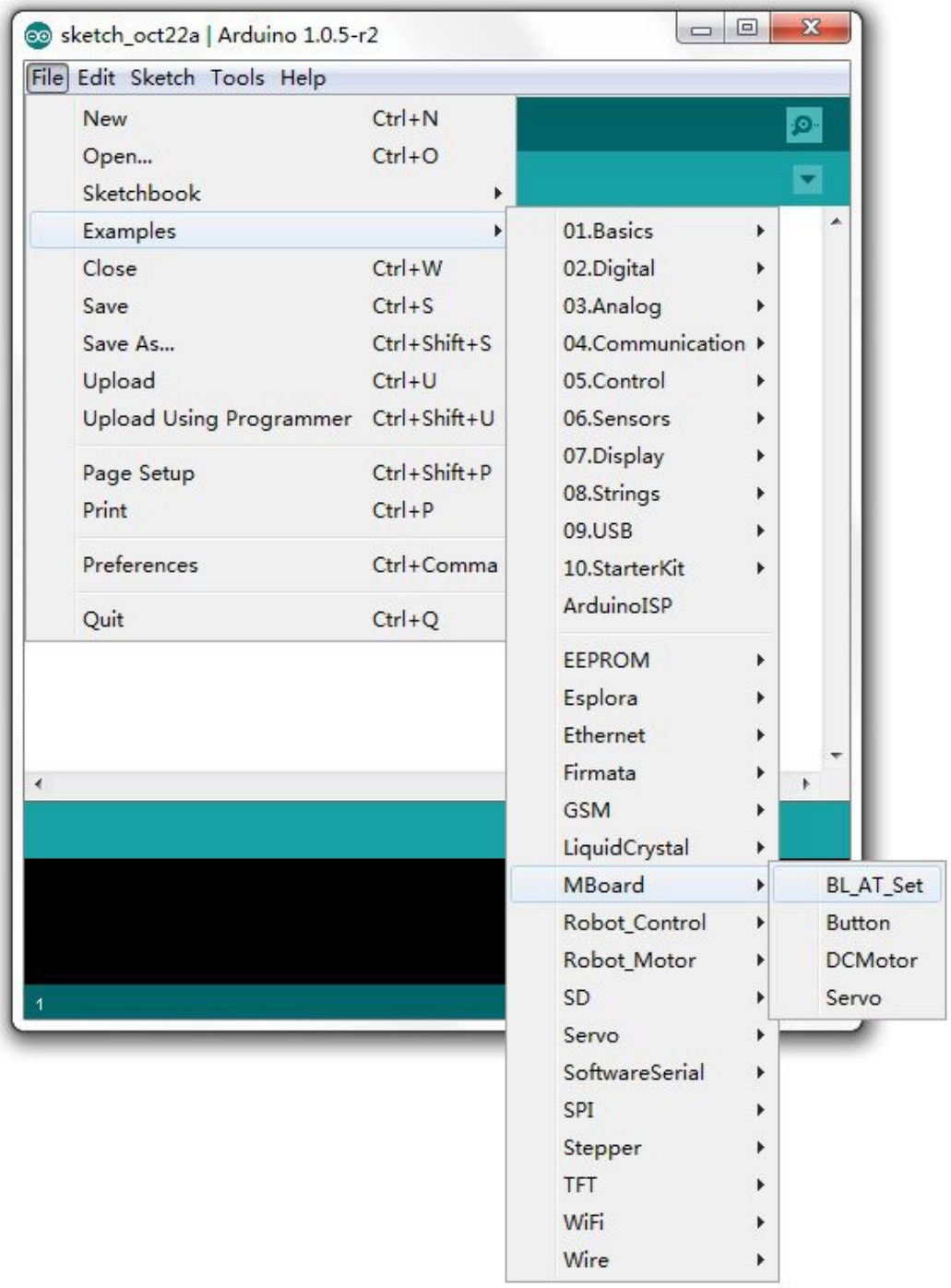
标识	颜色	功能
S1	白色	用户自定义按键（D7）
S2	红色	复位
S3	白色	蓝牙 4.0 记忆地址清除

3.8 跳线帽


M-Board Rev.A
Easy
State
Clr Addr
ERR
ISP
Battery
Servo
舵机电源和电机电源连同
舵机电源和电机电源断开
12V电机
6V电机

4 应用

在安装完驱动库后，重启IDE，在Examples里就会看到MBoard的应用例子了。



4.1 电机控制

打开 DCMotor 例子，下载程序到 M-Board，接上电池和电机。两路电机会同时正反转。

```
/*
  M-Board Library

 The circuit:
 * Motor1 AIN1 pin to digital pin 11
 * Motor1 AIN2 pin to digital pin 5
 * Motor2 BIN1 pin to digital pin 6
 * Motor2 BIN2 pin to digital pin 12

 Library originally added 12 August 2014
 by Blue

 This example code is in the public domain.

 http://4easy.cc
 */

#include <Easy.h>

int motorSpeed = 200;

DCMotor motor1(M1);
DCMotor motor2(M2);

void setup()
{

}

void loop()
{
    motor1.run(motorSpeed); // value: between -255 and 255.
    motor2.run(motorSpeed); // value: between -255 and 255.
    delay(2000);
    motor1.stop();
    motor2.stop();
    delay(100);
    motor1.run(-motorSpeed);
    motor2.run(-motorSpeed);
    delay(2000);
    motor1.stop();
    motor2.stop();
    delay(2000);
}
```

4.2 蓝牙通讯

打开 BL_AT_Set 例子，下载程序到 M-Board，电路板会设置蓝牙显示名称。注意背面标的蓝牙模块型号，是蓝牙 2.0，还是蓝牙 4.0。在代码中注释相应的语句。

```
void setup() {
 //Initialize serial and wait for port to open:
  Serial1.begin(9600);
  while (!Serial1) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  // prints title with ending line break
  Serial1.write("AT+NAMEtest");
}



void loop() {

}
```

（1）蓝牙 2.0 模块参数说明

进入AT 指令的方法：给模块上电，不配对的情况下，就是AT 模式了。指令间隔1s左右。

出厂参数：波特率9600N81，名字M-Board Rev.A-BL2.0，密码1234

1、测试通讯

发送：AT（返回OK，一秒左右发一次）

返回：OK

2、改蓝牙串口通讯波特率

发送：AT+BAUD1

返回：OK1200

例：发送：AT+BAUD2

返回：OK2400

……

1--------- 1200

2--------- 2400

3--------- 4800

4---------9600（默认就是这个设置）

5--------- 19200

6--------- 38400

7--------- 57600

8--------- 115200

9--------- 230400

A--------- 460800

B--------- 921600

C--------- 1382400

3、设置蓝牙显示名字

发送：AT+NAMEname

返回：OKsetname

参数name：所要设置的当前名称，即蓝牙被搜索到的名称。20 个字符以内。

例：发送AT+NAMEeasy

返回OKsetname

这时蓝牙名称改为easy

参数可以掉电保存，只需修改一次。PDA 端刷新服务可以看到更改后的蓝牙名称，名字不可超过20 个字符。

4、改蓝牙配对密码

发送：AT+PINxxxx

返回：OKsetPIN

参数xxxx：所要设置的配对密码，4 个数字。

例：发送：AT+PIN8888

返回：OKsetPIN

这时蓝牙配对密码改为8888，模块在出厂时的默认配对密码是1234。参数可以掉电保存，只需修改一次。

（2）蓝牙 4.0 模块的参数说明：

模块参数设置 AT指令

AT指令用来设置模块的参数，模块在未连线状态下可以进行AT指令操作，连线后进入串口透传模式。

模块启动大约需要150ms，所以最好在模块上电200ms以后才进行AT 指令操作。除特殊说明外，AT指令的参数设置立即生效。同时，参数和功能的修改，掉电不会丢失。

AT指令修改成功后统一返回 OK （“AT+RX、 AT+VERSION” 等查看信息类指令除外），不成功不返回任何信息。

指令集总

序号	AT 指令 (小写 x 表示参数)	作用	默认 状态	主/ 从 生效
1	AT	检测模块是否正常	-	M/S
2	AT+RX	查看模块基本参数	-	M/S
3	AT+DEFAULT	恢复出厂设置	-	M/S
4	AT+RESET	模块重启	-	M/S
5	AT+VERSION	获取模块版本、日期	-	M/S
6	AT+ROLE=x	主/从角色切换	S	M/S
7	AT+NAME=xxxxxxxxxxxx	修改蓝牙名称	M-Board Rev.A -BL4.0	M/S
8	AT+ADDR=xxxxxxxxxxxx	修改蓝牙地址	硬件地址	M/S
9	AT+RFPM=x	更改无线射频功率	0(4dBm)	M/S
10	AT+BAUD=x,y	修改串口波特率	9600,N	M/S
11	AT+CONT=x	是否可连接	0(可连)	M/S
12	AT+MODE=x	更改功耗模式	0	S
13	AT+AVDA=xxxxxxxxxxxx	更改广播数据	-	S
14	AT+TIME=x	组合工作模式 3 广播周期	5(s)	S

注：AT指令后面不用回车换行；如无特殊说明，本模块所有AT指令，一律不采用换行发送。

1、 测试指令

指令：AT

返回：OK。

2、 查看当前基本参数

显示蓝牙名称、主/从机、波特率、地址和密码等基本信息。

指令：AT+RX

返回：Name:M-Board Rev.A-BL4.0 -------->>>>蓝牙名是用户设定的名字

Role:Slave -------->>>>模块角色（主/从）

Baud:9600,NONE -------->>>>串口波特率，校验位

Addr:xx,xx,xx,xx,xx,xx -------->>>>蓝牙地址

PIN :000000 -------->>>>蓝牙密码（一般不需要密码）

3、 恢复出厂设置

指令：AT+DEFAULT

返回：OK

模块会自动重启，重启200ms后再进行新的操作

4、 模块重启指令

指令：AT+ RESET

返回：OK

模块会自动重启，请在模块重启 200ms后再进行新的操作

5、 查看软件版本指令

指令：AT+ VERSION

返回：HC-08V2.0，2014-08-22 （前面是软件版本，后面是发布日期）

6、 修改模块角色指令

设置指令：AT+ROLE=x

查询指令：AT+ROLE=？

x是模块角色代号，可设置为：M（主机）、S（从机）。

模块出厂默认是从机。

发送：AT+ROLE=M

返回：OK

模块设置为主机成功，模块自动重启后生效

发送：AT+ROLE=?

返回：Master

可以查看到模块角色是主机。

7、 修改蓝牙名称指令

设置指令：AT+ NAME=xxxxxxxxxxxx

查询指令：AT+ NAME=？

模块默认蓝牙名称是M-Board Rev.A-BL4.0，可以设置成其它名称（限12个字符以内,支持可视 ASCII码和部分转义字符。模块支持输入中文,安卓设备必须转换为“UTF8编码”才能够正常显示。发送超过12 个字符，则只认前面 12 个字符)。设置完成，模块自动重启后生效

例：

发送：AT+NAME=HCKJ

返回：OKsetNAME

发送：AT+NAME=?

返回：HCKJ

8、 修改蓝牙地址指令

设置指令：AT+ADDR=xxxxxxxxxxxx

查询指令：AT+ADDR=?

地址必须为12位的0~F大写字符，即16 进制字符。

例：

发送：AT+ADDR=1234567890AB

返回：OKsetADDR

设置完成，模块自动重启后生效

发送：AT+ADDR=？

返回：1234567890AB

发送：AT+ADDR=000000000000

返回：OKsetADDR

发送12个零，模块恢复成默认的硬件地址。模块出厂时默认使用硬件地址。

9、 修改射频功率指令

设置指令：AT+RFPM=x

查询指令：AT+RFPM=?

x是射频功率代号，如下表所示：

参数	射频发射功率
?	查看当前射频功率
0	4dBm（出厂默认值）
1	0dBm
2	-6dBm
3	-23dBm

例：

发送：AT+RFPM=2

返回：OK

模块射频功率修改成-6dBm，马上生效。

发送：AT+RFPM=?

返回：-6dBm

模块当前射频功率为-6dBm。

峰值电流超过 30mA（4dBm 时），射频功率最好设定为-6dBm 或者-23dBm。

10、 修改串口波特率指令

设置指令：AT+BAUD=x（只修改串口波特率）

AT+BAUD=x,y（修改串口波特率和校验位）

查询指令：AT+BAUD=?

x是串口波特率代号，y 是校验位代号，如下表所示：

参数	串口波特率x	参数	校验位
?	查看当前波特率	N	无校验
1200	1200bps	E	偶校验
2400	2400bps	O	奇校验
4800	4800bps	/	/
9600	9600bps(默认设置)	/	/
19200	19200bps	/	/
38400	34800bps	/	/
57600	57600bps	/	/
115200	115200bps	/	/

例：

发送：AT+BAUD=19200

返回：OK19200

模块串口波特率修改为19200bps，校验位和原来的一样

发送：AT+BAUD=4800,E

返回：OK4800,EVEN

模块串口波特率修改为4800bps，偶校验

发送：AT+BAUD=？

返回：4800,EVEN

显示模块当前串口波特率和校验位。

主机、从机透传通信时，9600bps 波特率以下每个数据包请不要超出 500 个字节，19200bps 波特率以上每个数据包的最大字节数请参考下表，数据包之间要有一定的时间间隔。下表是各种通信波特率下，时间间隔的参考值：

波特率	1200	2400	4800	9600	19200	38400	57600	115200
发 500 字节间隔时间(ms)	6800	3600	2000	1000	/	/	/	/
发 300 字节间隔时间(ms)	4200	2400	1200	600	400	/	/	/
发 100 字节间隔时间(ms)	1500	800	400	160	100	120	/	/
发 80 字节间隔时间(ms)	1000	650	320	120	80	60	100	/
发 60 字节间隔时间(ms)	800	500	250	100	60	60	60	100
发 20 字节间隔时间(ms)	200	100	50	20	20	20	20	20

注:

1、 以上是实测数据， 模块理论最快收发总速度： 2500 字节/秒， 建议把速度控制在 2000字节/秒。

2、建议每个数据包的字节数是 20 的整数倍。

3、模块发出的数据会自动分包为 20 字节的整数倍。就是发送一个100 字节数据包，在另外一个端上会收到多个数据包，每个数据包都是 20 的整数倍，总字节数为完整的 100字节。

4.3 舵机控制

打开 Servo 例子，下载程序到 M-Board，接上电池和舵机跳线帽，三路舵机会同时从 0 到 180 度来回转动。

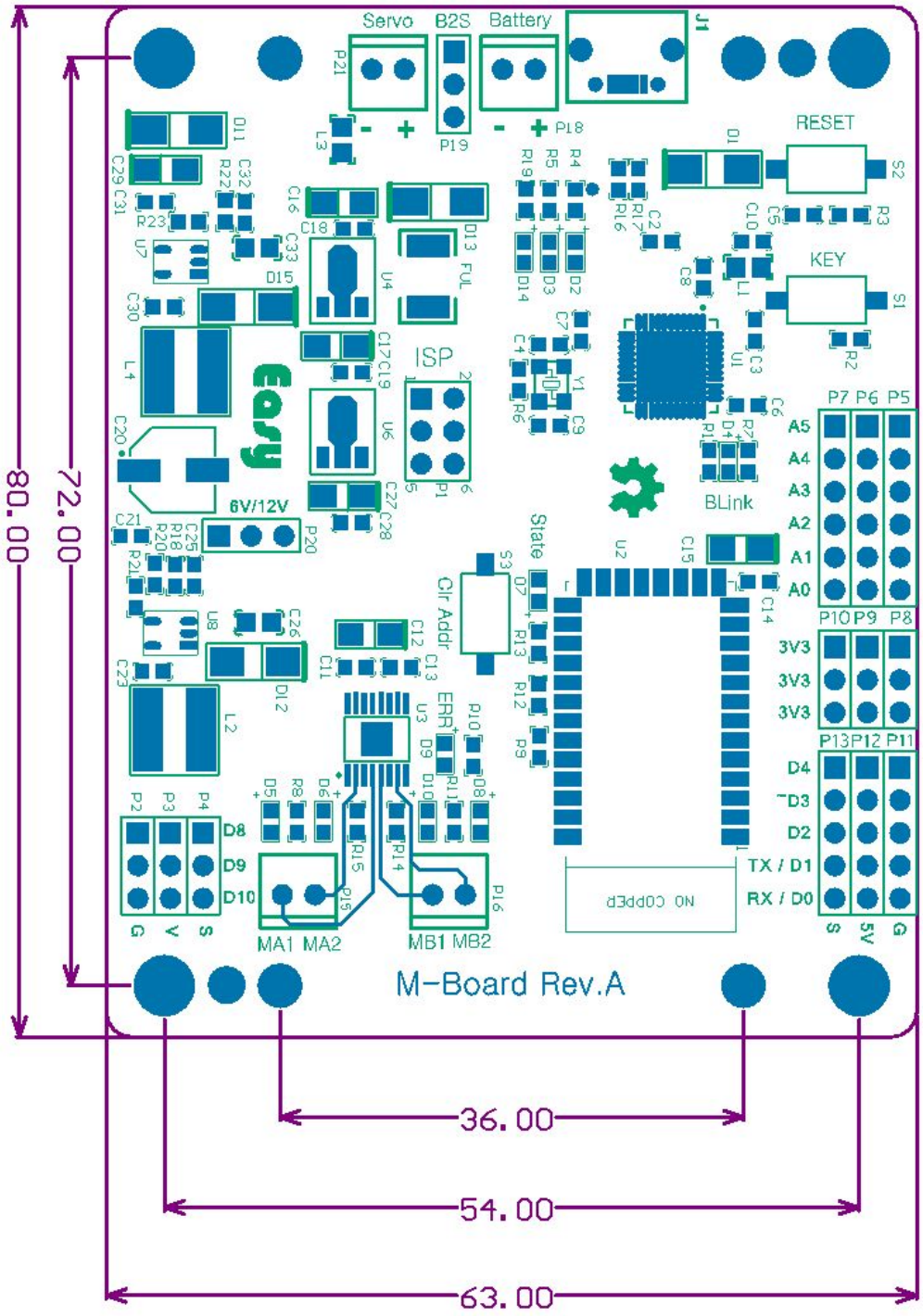
```
 * Servo1 Signal pin to digital pin 8
 * Servo2 Signal pin to digital pin 9
 * Servo3 Signal pin to digital pin 10

 Library originally added 12 August 2014
 by Blue

 This example code is in the public domain.

 http://4easy.cc
 */

#include <Servo.h>

Servo myservo1;  // create servo object to control a servo
                 // a maximum of eight servo objects can be created
Servo myservo2;
Servo myservo3;

int pos = 0;    // variable to store the servo position

void setup()
{
  myservo.attach(8);  // attaches the servo on pin 9 to the servo object
  myservo.attach(9);
  myservo.attach(10);
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo1.write(pos);     // tell servo to go to position in variable 'pos'
    myservo2.write(pos);
    myservo3.write(pos);
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees
  {
    myservo1.write(pos);              // tell servo to go to position in variable 'pos'
    myservo2.write(pos);
    myservo3.write(pos);
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

4.4 按键和 LED

打开 Button 例子，下载程序到 M-Board，按动按键能控制 LED 亮灭。

```
 * LED attached from pin 13 to ground
 * pushbutton attached to pin 7 from +5V
 * 10K resistor attached to pin 7 from ground

 * Note: on most Arduinos there is already an LED on the board
attached to pin 13.

Library originally added 12 August 2014
by Blue

This example code is in the public domain.

http://4easy.cc
 */

// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 7;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

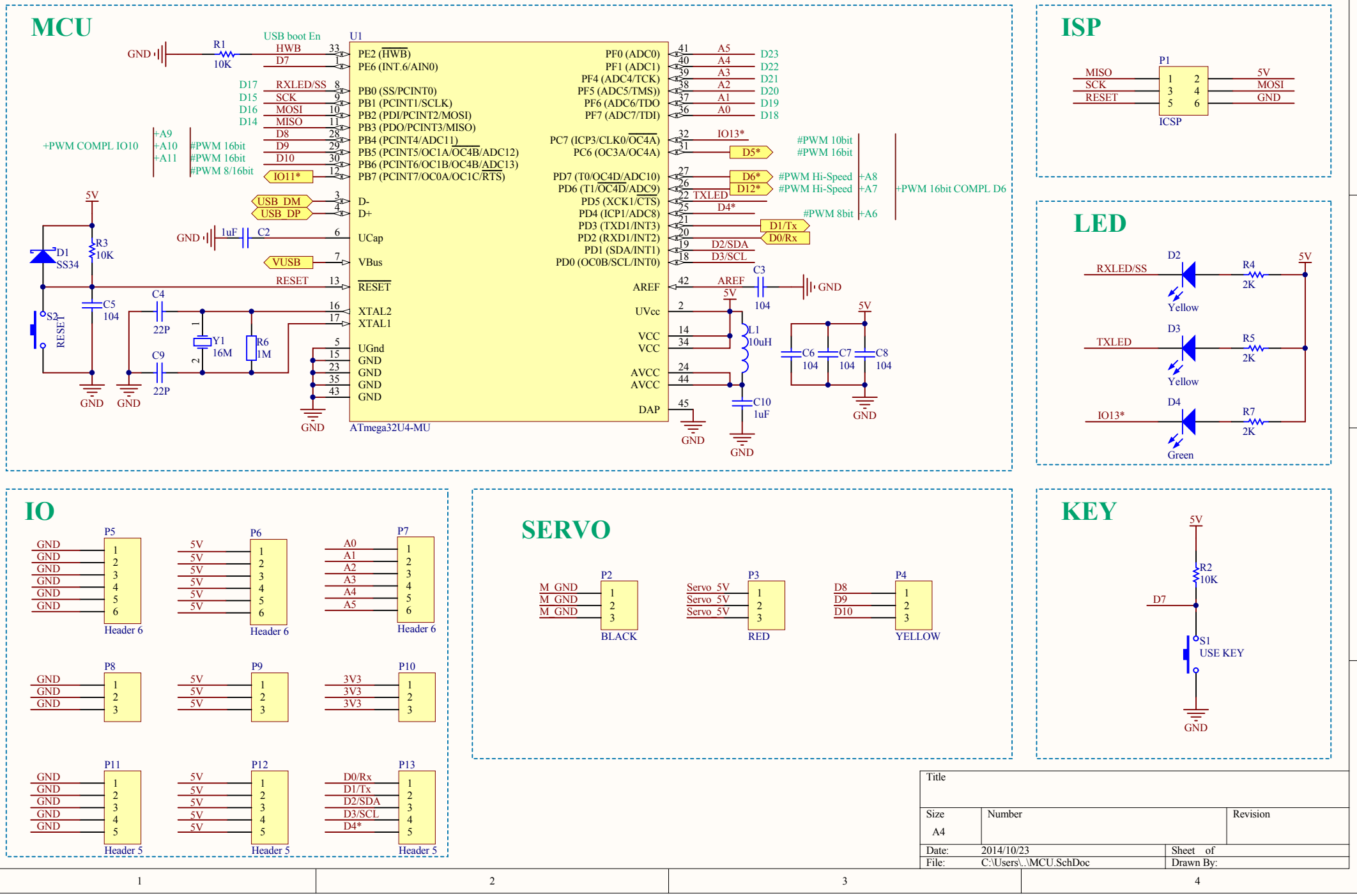
5 机械尺寸


Servo
B2S
Battery
RESET
KEY
ISP
BLink
Clr Addr
State
6V/12V
P7 P6 P5
A5
A4
A3
A2
A1
A0
P10 P9 P8
3V3
3V3
3V3
P13 P12 P11
D4
D3
D2
TX / D1
RX / D0
S 5V G
D8
D9
D10
G V S
MA1 MA2
MB1 MB2
M-Board Rev.A
80.00
72.00
36.00
54.00
63.00

6 版本历史

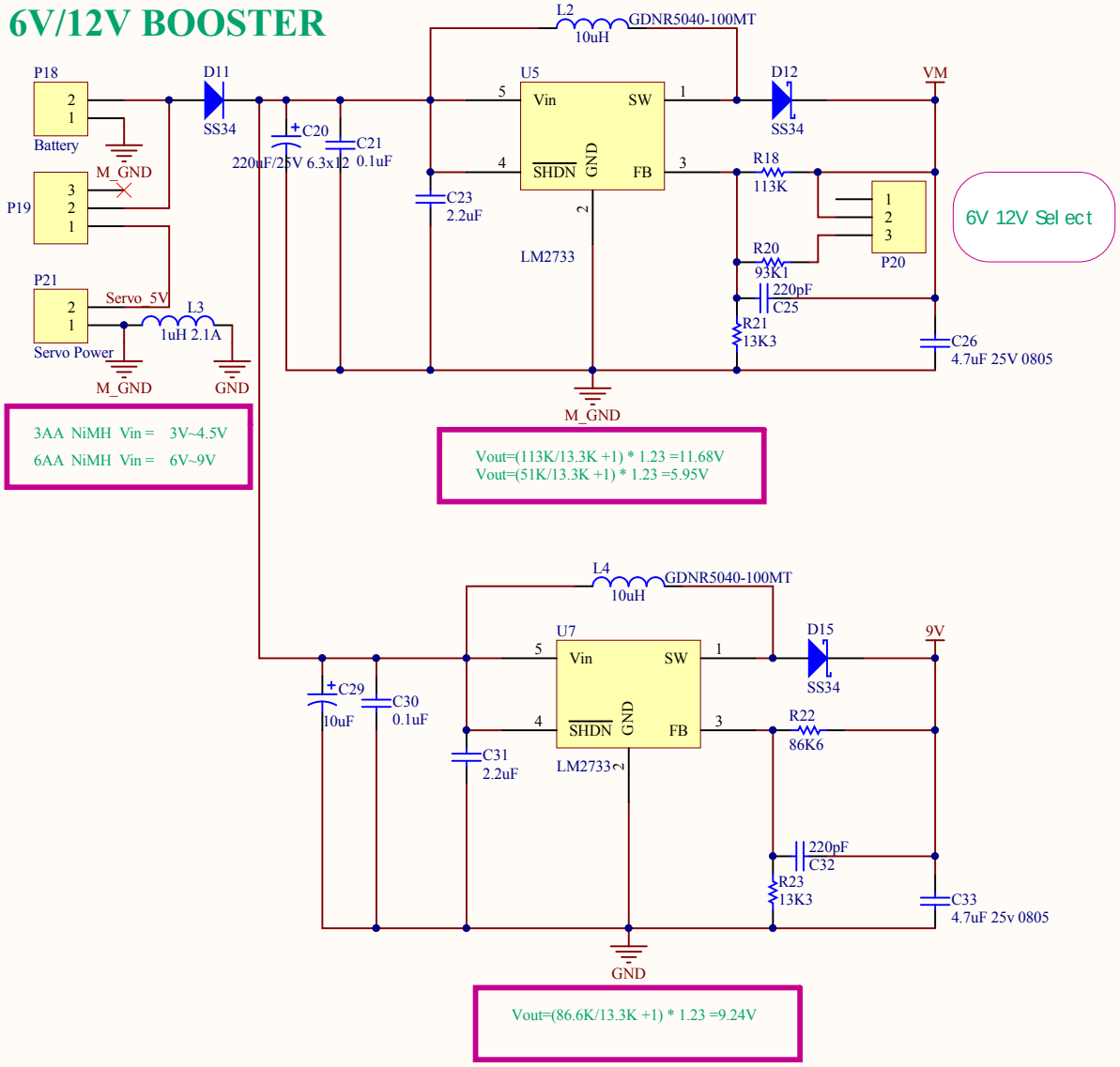
日期	版本	变化
22 October 2014	1.0	初稿

7 附录电路图

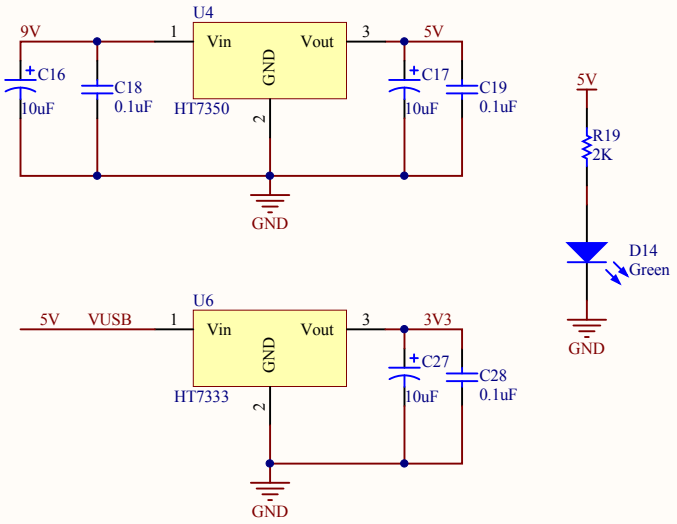

MCU
USB boot En
U1
R1
10K
HWB
D7
PE2 (HWB)
PE6 (INT.6/AIN0)
PB0 (SS/PCINT0)
PB1 (PCINT1/SCLK)
PB2 (PDI/PCINT2/MOSI)
PB3 (PDO/PCINT3/MISO)
PB4 (PCINT4/ADC11)
PB5 (PCINT5/OC1A/OC4B/ADC12)
PB6 (PCINT6/OC1B/OC4B/ADC13)
PB7 (PCINT7/OC0A/OC1C/RTS)
D17 RXLED/SS
D15 SCK
D16 MOSI
D14 MISO
D8
D9
D10
IO11*
+PWM COMPL IO10
+A9
+A10
+A11
#PWM 16bit
#PWM 16bit
#PWM 8/16bit
USB_DM
USB_DP
D-
D+
UCap
VBus
RESET
XTAL2
XTAL1
UGnd
GND
GND
GND
GND
PF0 (ADC0)
PF1 (ADC1)
PF4 (ADC4/TCK)
PF5 (ADC5/TMS))
PF6 (ADC6/TDO
PF7 (ADC7/TDI)
PC7 (ICP3/CLK0/OC4A)
PC6 (OC3A/OC4A)
PD7 (T0/OC4D/ADC10)
PD6 (T1/OC4D/ADC9)
PD5 (XCK1/CTS)
PD4 (ICP1/ADC8)
PD3 (TXD1/INT3)
PD2 (RXD1/INT2)
PD1 (SDA/INT1)
PD0 (OC0B/SCL/INT0)
A5 D23
A4 D22
A3 D21
A2 D20
A1 D19
A0 D18
IO13*
D5*
D6*
D12*
TXLED
D4*
D1/Tx
D0/Rx
D2/SDA
D3/SCL
#PWM 10bit
#PWM 16bit
#PWM Hi-Speed
#PWM Hi-Speed
#PWM 8bit
+A8
+A7
+A6
+PWM 16bit COMPL D6
AREF
UVcc
VCC
VCC
AVCC
AVCC
DAP
ATmega32U4-MU
5V
D1
SS34
R3
10K
C5
104
S2
RESET
C4
22P
C9
22P
Y1
16M
R6
1M
1uF
C2
C3
104
L1
10uH
C6
104
C7
104
C8
104
C10
1uF
GND
ISP
P1
MISO
SCK
RESET
5V
MOSI
GND
ICSP
LED
D2
RXLED/SS
R4
2K
Yellow
D3
TXLED
R5
2K
Yellow
D4
IO13*
R7
2K
Green
IO
P5
Header 6
P6
Header 6
P7
A0
A1
A2
A3
A4
A5
Header 6
P8
P9
P10
3V3
Header 5
P11
P12
Header 5
P13
D0/Rx
D1/Tx
D2/SDA
D3/SCL
D4*
Header 5
SERVO
P2
M_GND
BLACK
P3
Servo_5V
RED
P4
YELLOW
KEY
R2
10K
S1
USE KEY
Title
Size
A4
Number
Revision
Date:
2014/10/23
Sheet of
File:
C:\Users\..\MCU.SchDoc
Drawn By:

System Power

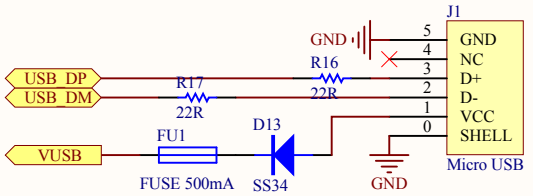
6V/12V BOOSTER



3.3V 5V REGULATOR

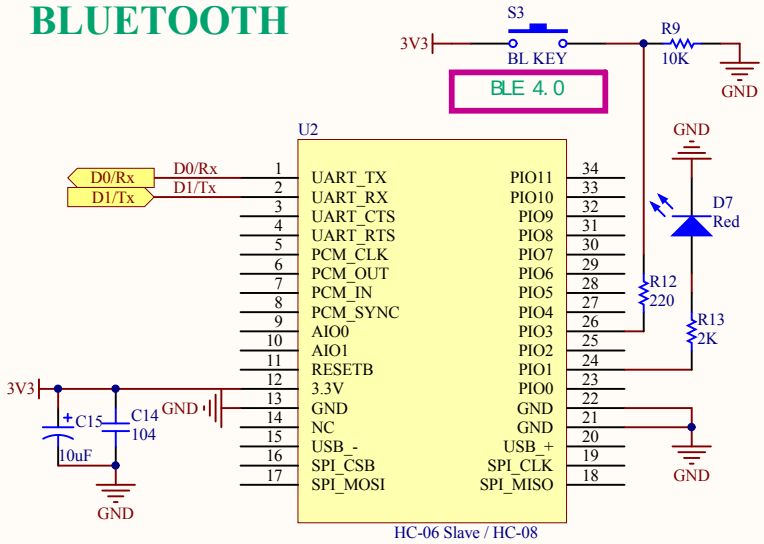


USB



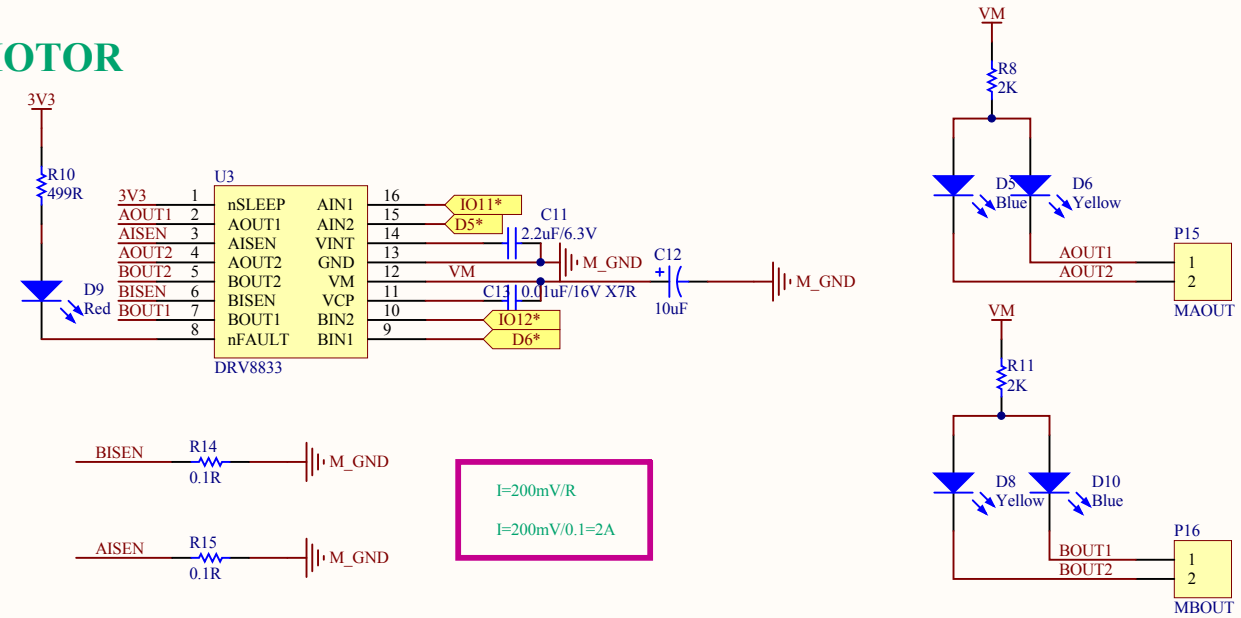
Title		
Size: A4	Number	Revision
Date: 2014/10/23		Sheet of
File: C:\Users\..\Power.SchDoc		Drawn By:

BLUETOOTH



Title		
Size: A4	Number	Revision
Date: 2014/10/23		Sheet of
File: C:\Users\..\BLUETOOTH.SchDoc		Drawn By:

MOTOR



Title		
Size A4	Number	Revision
Date: 2014/10/23		Sheet of
File: C:\Users\..\MOTOR.SchDoc		Drawn By: